REMARKS

Claims 1-9 and 11-71 are pending, of which claims 1, 25, 48, 54, 55, 56, and 61, are independent. In the Office Action, the Examiner rejects claims 1-9 and 11-71. Applicants amend claims 1, 11, 13, 15, 25, 34-35, 37, 39, 40, 48, 50-56, 61-64, and 70-71 herein, and add new claim 72. No new matter is added by these amendments. Applicants respectfully request reconsideration of the outstanding rejections and passage of the claims to allowance in light of the following.

New claim 72 recites subject matter similar to the subject matter of originally filed claim 10. Claim 10 was rejected in the June 4, 2008 Office Action under 35 U.S.C. §112 for reciting "another block." Claim 72 recites "a second block" instead of "another block."

The claims are amended to recite ***block methods*** rather than *execution methods*. Support for this amendment can be found throughout the Specification as originally filed. For example, support may be found in the Specification at page 25, line 6 through page 26, line 28.

I. Rejections under 35 U.S.C. §112

The Examiner rejects claims 1-9 and 11-71 under 35 U.S.C. §112 as failing to comply with the written description requirement. The Examiner asserts that Applicants' amendments present new matter that was not disclosed in the specification. In particular, the Examiner states that "the concurrency of the debugger and its interface with the model being executed, and the list view, are not disclosed adequately" in the specification (Office Action at page 4, paragraph 9.1).

In order to expedite prosecution, Applicants amend the independent claims to recite ***a graphical debugger that interfaces with ...*** Support for this feature of claim 1 can be found throughout the application as filed. For example, support can be found at page 5, last paragraph. Further support can be found at page 8, lines 10-20; at page 38, line 21 through page 39, line 15; and at Figure 15.

The dependent claims are rejected "by virtue of their dependency" (Office Action at page 4, paragraph 12). In light of the above, Applicants respectfully request that the Examiner reconsider and withdraw the 35 U.S.C. §112 rejection of claims 1-9 and 11-71.

II. Rejections under 35 U.S.C. §103(a)

A. Claims 1-3, 5-9, 17-22, 24-27, 29-33, 41-46, and 48-71

In the Office Action, claims 1-3, 5-9, 17-22, 24, 25-27, 29-33, 41-46, and 48-71 were rejected under 35 U.S.C. §103(a) as being unpatentable over MathWorks Simulink® "Dynamic System Simulation for MATLAB," 1997 (hereinafter "MathWorks") in view of Official Notice taken, and further in view of U.S. Patent No. 7,107,578 to Alpern (hereafter "Alpern"). Applicants respectfully traverse the rejection.

1. Claims 1-3, 5-9, 17-22, 24, and 70

Applicants' claim 1 recites,

> A method comprising the steps of:
> providing a graphical debugger that interfaces with:
> a model view of a model being executed, and
> ***a block method execution list view of block methods called during execution of said model***,
> said model comprising a block that includes a plurality of block methods, said graphical debugger having debug information related to the execution of said model, ***said debug information indicating an order of execution of said plurality of block methods for said block and a start time or a stop time of said plurality of block methods for said block that are executed during the execution of said model***; and
> outputting said debug information to a user, ***said debug information allowing the user to determine proper or improper operation for at least a subset of said plurality of block methods that are executed in said block during the execution of said model***.

MathWorks, the Official Notice taken, and Alpern, alone or in any reasonable combination, do not disclose or suggest ***a block method execution list view of block methods called during execution of said model***, nor do the cited references disclose or suggest ***said debug information indicating an order of execution of said plurality of block methods for said block and a start time or a stop time of said plurality of block methods for said block that are executed during the execution of said model***, nor do the cited references disclose or suggest ***said debug information allowing the user to determine proper or improper operation for at least a subset of said plurality of block methods that are executed in said block during the execution of said model*** as present in claim 1.

In order to expedite prosecution, Applicants amend independent claim 1 to clarify that

the methods of claim 1 are block methods. For example, claim 1 recites ***a block method execution list view of block methods called during execution of said model***. Such a block method execution list view is not disclosed or suggested in any of the cited references.

A block represents a unit of functionality in a model (Specification at page 3, first full paragraph). Block methods provide a means of implementing the functionality of the block. In addition to block methods, blocks may contain other information as well, such as numerical information (Specification at page 14, first full paragraph, and page 15, line 32, through page 17, line 4).

The Examiner asserts that the MathWorks reference discloses an execution list view at page 3-49 (Office Action at page 4). However, the cited passage of the MathWorks reference does not address ***block methods***. Instead, the MathWorks reference explicitly states that the list view shown on page 3-49 depicts a list of blocks, and not block methods (MathWorks at page 3-49, stating that "the blocks list" contains "the names of blocks in the selected system"). Applicants are very familiar with the described software, as the Applicants' employer, the MathWorks, Inc., designed the software in question and published the cited reference. The version of Simulink described in the cited MathWorks reference does not include ***a block method execution list view of block methods called during execution of said model***, as recited in claim 1.

Alpern also does not disclose or suggest ***a block method execution list view of block methods called during execution of said model***. Alpern is generally directed to debugging a computer program that is executed by multiple virtual machines (Alpern at col. 1, lines 15-20). Alpern allows a user to the view modules of code, but these modules of code do not correspond to ***a block method execution list view of block methods called during execution of said model***

The Examiner suggests that the modules of code in Alpern could be considered to be equivalent to blocks in a block diagram model, and that the code in Alpern could be considered to be block methods (Office Action at page 3, paragraph 7.1). Applicants respectfully disagree. As noted above, a block represents a unit of functionality in a system, and may include a collection of methods that are invoked by the execution engine (Specification at page 3, first full paragraph). The block methods provide a means of implementing the functionality of the block. In addition to block methods, blocks may contain other information as well (Specification at

page 14, first full paragraph, and page 15, line 32, through page 17, line 4). Accordingly, it is not reasonable to treat a block as the equivalent of a mere "piece of code." Indeed, in addition to block methods, blocks may also include other elements such as numerical data, and "not being able to see the relevant numerical data superimposed on the graphical topology as captured by the block diagram of the model during debugging is a major impediment to efficient debugging" (Specification at page 4).

The routines of Alpern cannot represent the blocks, as suggested by the Examiner at page 3 of the Office Action, because the "routines" of Alpern fail to take into account many aspects of the "elemental dynamic system" represented by each block (Specification at page 14, first full paragraph, and page 15, line 32, through page 17, line 4). Because the blocks do not merely represent "pieces of code," as asserted by the Examiner, the routines of Alpern do not represent blocks or block methods.

Indeed, the type of code that Alpern operates on is explicitly considered in the present Specification (see, e.g., Specification at pages 4 and 26). For example, Alpern's debugger may operate on modules of code written in Java or SQL. Similarly a block diagram engine in the present Specification may generate "instructions in a high-level software language such as C, C++, Ada, etc" (Specification at page 26). However, the Specification notes that this high-level code also does not correspond to block methods. Indeed, the present Specification notes that, instead of corresponding to the methods performed by blocks, generated code like the kind processed by Alpern does not correspond 1-to-1 with block diagram methods because the generated code is translated from the original block diagram model and optimized (Specification at page 4). During code generation, some elements of the model may be removed, some elements may be added that were not present in the model, and the code may be further modified to facilitate execution (see, e.g., Specification at page 4). After these steps have been carried out, the code no longer represents ***block methods***, as recited in claim 1. Accordingly, the view of routines shown, for example, at Figure 5 of Alpern is not ***a block method execution list view of block methods called during execution of said model***

The Official Notice taken does not concern ***a block method execution list view of block methods called during execution of said model***. The Official Notice taken states that "a block includes a plurality of execution methods" (Office Action at page 5, lines 5-9).

Accordingly, none of Alpern, MathWorks, or the Official Notice taken disclose or suggest debugging at the block-method level. Therefore, none of the cited references disclose or suggest ***a block method execution list view of block methods called during execution of said model***.

In light of the above discussion, the cited references also do not disclose or suggest ***said debug information indicating an order of execution of said plurality of block methods for said block and a start time or a stop time of said plurality of block methods for said block that are executed during the execution of said model***. The debugger in the MathWorks reference provides debug information indicating information about the blocks, and does not indicate ***an order of execution of said plurality of block methods for said block*** nor ***a start time or a stop time of said plurality of block methods***. Alpern may provide debug information about methods called during the execution of generated code, but the code of Alpern does not correspond to ***block methods*** for the reasons discussed above. The Official Notice taken does not address ***debug information*** at all.

Further, the cited references do not disclose or suggest ***said debug information allowing the user to determine proper or improper operation for at least a subset of said plurality of block methods that are executed in said block during the execution of said model*** as present in claim 1. For the reasons discussed above, none of the cited references disclose or suggest a proper or improper operation for any of a ***plurality of block methods***.

Further, one having ordinary skill in the art could not combine Alpern's debugger with the block diagrams of the MathWorks reference. A reasonable expectation of success is required in order to combine references (MPEP at §2143.02). One having ordinary skill in the art would not have a reasonable expectation of success in combining the conventional debugger of Alpern with the block diagrams of MathWorks.

Alpern relies on associations between the high-level code and the compiled code in order to operate the debugger (Alpern at col. 8, line 66 – col. 9, line 7). However, as described in the Specification, the block diagram as designed by a user may be modified as part of compilation before it is simulated (Specification at page 4). The compilation process for a block diagram may add or remove elements to optimize simulation (Specification at page 4). This process includes steps not contemplated by Alpern and would eliminate the associations that Alpern

relies on in order to debug software modules.

In the present specification, support for the debugging of ***block methods*** is incorporated into the compiler and execution engine, as described in detail in the specification at page 23, line 15 *et seq*. For example, the instant compiler may transform the graphical block diagram into a compiled directed graph consisting of arcs and vertices, but nothing in Alpern would suggest to one of ordinary skill in the art to perform such a step. Without functionality like the compiler design described in the instant specification, a conventional debugger would not provide a user with meaningful information about the ***block methods***.

Because Alpern uses traditional methodologies, Alpern's debugger would not be capable of interpreting the ***block methods*** of the present claims. If Alpern's debugger was applied to interpret code compiled from a block diagram, the debugger would step through the code as generated by the compiler, but the code would not "line up" with the ***block methods*** (due to the modification of the block diagram at compile time). Instead of reflecting the methods for each block, the code referenced by the debugger would reflect "simulation information embodied by the compiled version of the block diagram" (Specification at page 4).

The information returned by Alpern's conventional debugger would be meaningless to a user debugging ***block methods***. Because one having ordinary skill in the art would not have a reasonable expectation of success in combining Alpern and MathWorks, Alpern and MathWorks may not be combined (MPEP at §2143.02.II).

For at least the reasons given above, MathWorks, the Official Notice taken, and Alpern, alone or in any reasonable combination, do not disclose or suggest each and every element of independent claim 1. Claims 2-3, 5-9, 17-22, 24, and 70 depend from claim 1 and, as such, include each and every element of claim 1. Thus, MathWorks, the Official Notice taken, and Alpern, alone or in any reasonable combination, do not disclose or suggest each and every element of claims 2-3, 5-9, 17-22, 24, and 70. Therefore, Applicants respectfully request that the 35 U.S.C. §103(a) rejection of claims 1-3, 5-9, 17-22, 24, and 70 be withdrawn.

2. Claims 25-27, 29-33, 41-46, and 71

Applicants' claim 25 recites,

> 25. A medium for use in a modeling and execution environment on an electronic device, said medium holding executable instructions on the electronic device for

performing an execution method, said method comprising the steps of:
providing a graphical debugger that interfaces with:
a model view of a model being executed, and
a block method execution list view of block methods called during an execution of the model,
said model comprising a block that includes a plurality of block methods, said graphical debugger having debug information related to the execution of said model, ***said debug information indicating at least one of the order of the execution of a plurality of block methods in said model and a start time or a stop time of at least one block method executed during the execution of said model***; and
outputting said debug information to a user, ***said debug information allowing the user to determine proper or improper operation for at least a subset of said plurality of block methods for the block that are executed during the execution of said model***.

Applicants respectfully submit that MathWorks, the Official Notice taken, and Alpern, alone or in any reasonable combination, do not disclose or suggest ***a block method execution list view of block methods***, nor ***indicating at least one of the order of the execution of a plurality of block methods in said model and a start time or a stop time of at least one block method***, nor ***proper or improper operation for at least a subset of said plurality of block methods,*** as recited in claim 25.

As discussed above with respect to claim 1, MathWorks provides information at the block level, while Alpern provides information at the level of generated code. The Official Notice taken does not concern block methods at all.

Claims 26-27, 29-33, 41-46, and 71 depend from claim 25, and thus include each and every element of claim 25. In light of the above arguments, Applicants respectfully request that the 35 U.S.C. §103(a) rejection of claims 25-27, 29-33, 41-46, and 71 be withdrawn.

3. Claims 48-53

Applicants' claim 48 recites,

48. A computer-implemented method, comprising:
identifying a first block method operating in a first environment of a computer-based modeling application that executes a model, where the first environment is one of a text-based environment, a time-based block diagram, a state based block diagram, or a data-flow diagram;
identifying a second block method operating in a second environment, where the second environment differs from the first environment;

> ***debugging the first block method and the second block method with a debugger that is interfaced with the first environment and the second environment while the computer-based model operates on behalf of a user***; and
> generating output information for the user or for a destination, ***the output information identifying when the first block method or the second block method are operating, identifying an operation performed by the first block method or the second block method at a determined location in the first block method or the second execution method, or identifying an error related to the first block method or the second block method during execution of the computer-based model.***

Applicants respectfully submit that MathWorks, the Official Notice taken, and Alpern, alone or in any reasonable combination, do not disclose or suggest any steps performed at the level of a ***block method***, as recited in claim 48.

As discussed above with respect to claim 1, MathWorks provides information at the block level, while Alpern provides information at the level of generated code. The Official Notice taken does not concern debugging block methods.

Therefore, the cited references do not disclose or suggest at least ***identifying a first block method operating in a first environment of a computer-based modeling application that executes a model***, ***identifying a second block method operating in a second environment***, ***debugging the first block method and the second block method with a debugger that is interfaced with the first environment and the second environment while the computer-based model operates on behalf of a user***, and ***the output information identifying when the first block method or the second block method are operating, identifying an operation performed by the first block method or the second block method at a determined location in the first block method or the second execution method, or identifying an error related to the first block method or the second block method during execution of the computer-based model***, as recited in claim 48.

Further, Alpern does not allow an execution method that ***operates in a first environment of a computer-based modeling application that executes a model*** to be debugged, and one having ordinary skill in the art would not have a reasonable expectation of success in implementing the operations of Alpern in ***a computer-based modeling application that executes a mode***. Alpern is concerned with debugging modules for virtual machines that were written in different languages. There is no indication that the methods Alpern utilizes would allow

Alpern's debugging method to be applied in ***a first environment of a computer-based modeling application that executes a model***. As noted above in relation to claim 1, there are a number of difficulties with debugging in such an environment which Alpern does not address.

Therefore, MathWorks, the Official Notice taken, and Alpern, alone or in any reasonable combination, do not disclose or suggest each and every element of independent claim 48. Claims 49-53 depend from claim 48, and thus include each and every element of claim 48. In light of the above arguments, Applicants respectfully request that the 35 U.S.C. §103(a) rejection of claims 48-53 be withdrawn.

4. Claim 54

At the outset, Applicants note (with reference to page 25 of the December 23, 2008 Response) that the Examiner continues to improperly reject independent claim 54. As discussed in more detail below (and as was discussed in the previous Response), claim 54 includes a number of limitations not recited in claim 1; however, the Examiner continues to reject claim 54 as including "substantially same limitations (sic)" as claim 1. The Examiner has not considered a number of features of claim 54 which are not present in claim 1, and thus has failed to establish a *prima facie* case of obviousness.

Claim 54 recites,

> 54. A method, comprising:
> receiving information about a first block method and a second block method on behalf of a graphical model comprising blocks, where at least one of the blocks includes the first block method and at least one other block method or the second block method and the at least one other block method, where the first block method or the second block method are related to one or more of the blocks;
> identifying at least a portion of the first block method or the second block method when the first block method or the second block method are running, respectively;
> obtaining information about the running of the first block method or the second block method using the identifying; and
> providing debugging information to a user via a display or providing debugging information to a destination device, the debugging information provided by a debugger that interfaces with the graphical model and a list of block methods called during simulation of the graphical model, the debugging information identifying the first block method or the second block method and information about the first block method or the second block method, respectively.

The Examiner states, "as per claim 54, note the rejection of claim 1 above. The Instant Claim recites substantially same (sic) limitations as the above-rejected claims and therefore rejected (sic) under same prior-art teachings" (Office Action at page 8). However, claims 1 and 54 recite very different limitations. The Examiner has not considered claim 54 and improperly rejects the claim.

For example, MathWorks, the Official Notice taken, and Alpern, alone or in any reasonable combination, do not disclose or suggest at least ***receiving information about a first block method and a second block method on behalf of a graphical model comprising blocks, where at least one of the blocks includes the first block method and at least one other block method or the second block method and the at least one other block method***, as present in claim 54. None of the above-cited references receive information about execution methods on behalf of a graphical model.

Additionally, as noted above with respect to claim 1, MathWorks and the Official Notice taken are not concerned with ***block methods*** included in blocks, which are recited throughout claim 54.

Further, as noted above, Alpern does not address a ***graphical model comprising blocks***, and there is no indication that Alpern could be modified to work with such a graphical model. One having ordinary skill in the art would not have a reasonable expectation of success in applying Alpern's conventional debugger in a block-method-level debugging situation.

Therefore, MathWorks, the Official Notice taken, and Alpern, alone or in any reasonable combination, do not disclose each and every element of claim 54. Applicants respectfully request that the 35 U.S.C. §103(a) rejection of claim 54 be withdrawn.

5. Claim 55

Claim 55 recites,

55. A method, comprising:
identifying a first root method comprising one or more child methods, the first root method related to a graphical modeling application;
identifying a second root method related to the graphical modeling

application;

running the first root method and the second root method in a graphical debugger to obtain information about the operation of the first root method or the second root method, ***the graphical debugger interfaced with a method list of block methods called during simulation of a model in the graphical modeling application and a model view of a model in the graphical modeling application***; and

displaying a debugging result to a destination, the debugging result comprising visual identifiers related to the operation of the first root method, the one or more child methods or the second root method, error information about the first root method, the one or more child methods or the second root method, an execution result for the first root method, the one or more child methods or the second root method, or status information related to the first root method, the one or more child methods or the second root method.

Claim 55 recites that ***the graphical debugger is interfaced with a method list of block methods called during simulation of a model in the graphical modeling application and a model view of a model in the graphical modeling application***. Applicants respectfully submit that MathWorks, the Official Notice taken, and Alpern, alone or in any reasonable combination, do not disclose or suggest the above-quoted feature of amended independent claim 55. None of the cited references is concerned with a ***method list of block methods***.

Further, MathWorks, the Official Notice taken, and Alpern, alone or in any reasonable combination, do not disclose or suggest ***the debugging result comprising visual identifiers related to the operation of the first root method, the one or more child methods or the second root method, error information about the first root method, the one or more child methods or the second root method, an execution result for the first root method, the one or more child methods or the second root method, or status information related to the first root method, the one or more child methods or the second root method***, as present in claim 55.

MathWorks does not address debugging root methods, because MathWorks is focused at the block level and not the level of root methods. Thus, MathWorks does not give a debugging result with visual identifiers related to the operation of root or child methods (see, e.g., MathWorks at page 12-16). MathWorks does not disclose or suggest displaying status information related to root or child methods, but can display status information related to blocks and overall system states (MathWorks at pages 12-12, 12-14, and 12-16). The Official Notice taken is also not concerned with root methods, and so does not disclose or suggest the above-

cited feature of claim 55.

Alpern also does not disclose or suggest ***the debugging result comprising visual identifiers related to the operation of the first root method, the one or more child methods or the second root method, error information about the first root method, the one or more child methods or the second root method, an execution result for the first root method, the one or more child methods or the second root method, or status information related to the first root method, the one or more child methods or the second root method***. As noted in a preceding clause in claim 55, the first and second root methods are ***related to a graphical modeling application***. Alpern does not suggest a way for a debugging method to be applied to a graphical modeling application.

Thus, MathWorks, the Official Notice taken, and Alpern, alone or in any reasonable combination, do not disclose or suggest all the elements of claim 55. Therefore, Applicants respectfully request that the 35 U.S.C. §103(a) rejection of claim 55 be withdrawn.

6. Claims 56-60

Claim 56 recites,

> 56. A method for implementing a user interface for debugging a graphical model, the method comprising:
>
> ***displaying a hierarchy comprising information about a first root method, one or more child methods related to the first root method, or a second root method, the hierarchy displaying information about the first root method, the one or more child methods, or the second root method in an arrangement representing a relationship among the first root method, the one or more child methods, or the second root method***; and
>
> displaying an indicator on the hierarchy proximate to the first root method, the one or more child methods, or the second root method, the indicator denoting a status of the first root method, the one or more child methods, or the second root method, where the status indicates whether the first root method, the one or more child methods, or the second root method are operating according to determined parameters, ***the determination of whether the first root method, the one or more child methods, or the second root method are operating according to determined parameters made by a debugger that interfaces with a method list of block methods called during simulation of the graphical model a model view of the graphical model***.

Claim 56 recites that ***the determination of whether the first root method, the one or more child methods, or the second root method are operating according to determined***

parameters is made by a debugger that interfaces with a method list of block methods called during simulation of the graphical model a model view of the graphical model. Applicants respectfully submit that MathWorks, the Official Notice taken, and Alpern, alone or in any reasonable combination, do not disclose or suggest the above-quoted feature of amended independent claim 56. As noted above with respect to claim 55, none of the cited references are concerned with a method list ***of block methods***.

Further, MathWorks, the Official Notice taken, and Alpern, alone or in any reasonable combination, do not disclose or suggest ***displaying a hierarchy comprising information about a first root method, one or more child methods related to the first root method, or a second root method, the hierarchy displaying information about the first root method, the one or more child methods, or the second root method in an arrangement representing a relationship among the first root method, the one or more child methods, or the second root method***, as present in claim 56.

MathWorks and the Official Notice taken are concerned with block-level debugging. Instead of displaying a hierarchy of root and child methods, MathWorks displays a block execution order, and not information relating to root and child methods. For instance, compare MathWorks at page 12-16, showing the hierarchy of MathWorks, with Figure 18A of the Application, which depicts the parent-child hierarchy. The block execution order of MathWorks is silent as to root or child methods.

Moreover, Alpern does not address root and child methods as described in claim 56. In fact, the frames and execution methods that the Examiner cites in Alpern come from *different modules programmed in different languages* (Alpern at col. 11, lines 14-23). Because they are programmed separately and in different languages, there is no root or child method relationship between the different frames in Figure 5 of Alpern.

Thus, MathWorks, the Official Notice taken, and Alpern, alone or in any reasonable combination, do not disclose or suggest all the elements of claim 56. Claims 57-60 depend from claim 56, and thus include ever element of claim 56. In light of the above arguments, Applicants respectfully request that the 35 U.S.C. §103(a) rejection of claims 56-60 be withdrawn.

7. Claims 61-69

Claim 61 recites,

> 61. A method for debugging operation of a graphical icon, the method comprising:
>
> ***identifying a plurality of block methods for the graphical icon using a plurality of regions related to the graphical icon***;
>
> ***displaying information about a first one of the plurality of block methods in a first one of the plurality of regions or information about a second one of the plurality of block methods in a second one of the plurality of regions***; and
>
> associating the information in the first one of the plurality of regions or information in the second one of the plurality of regions with a graphical debugger to provide a user with debugging results for the first one of the plurality of block methods or the second one of the plurality of block methods, the debugging results allowing the user to identify desirable operations performed on behalf of the graphical icon or undesirable operations performed on behalf of the graphical icon, ***the graphical debugger interfacing with a view of the graphical icon and an execution methods list of block methods performed on behalf of the graphical icon***.

Claim 61 recites ***identifying a plurality of block methods for the graphical icon using a plurality of regions related to the graphical icon***, ***displaying information about a first one of the plurality of block methods in a first one of the plurality of regions or information about a second one of the plurality of block methods in a second one of the plurality of regions***, and ***the graphical debugger interfacing with a view of the graphical icon and an execution methods list of block methods performed on behalf of the graphical icon***. As noted above with respect to claim 1, the cited references do not disclose or suggest debugging block methods. Applicants respectfully submit that MathWorks, the Official Notice taken, and Alpern, alone or in any reasonable combination, do not disclose or suggest the above-quoted feature of amended independent claim 61.

Thus, MathWorks, the Official Notice taken, and Alpern, alone or in any reasonable combination, do not disclose or suggest all the elements of claim 61. Claims 62-69 depend from claim 61, and thus include ever element of claim 61. In light of the above arguments, Applicants respectfully request that the 35 U.S.C. §103(a) rejection of claims 61-69 be withdrawn.

B. Claims 4, 10-16, 23, 28, 34-40, and 47

Claims 4, 10-16, 23, 28, 34-40, and 47 were rejected under 35 U.S.C. §103(a) as being

unpatentable over MathWorks in view of the Official Notice taken and Alpern, and further in view of Fenlason's "GNU gprof" (1998). Applicants respectfully traverse the rejection.

Claims 4, 10-16, and 23 depend from claim 1 and, as such, include each and every element of claim 1. Claims 28, 34-40, and 47 depend from claim 25 and, as such, include each and every element of claim 25. As noted above, MathWorks, the Official Notice taken, and Alpern, alone or in any reasonable combination, do not disclose or suggest each and every element of claims 1 and 25.

The addition of GNU gprof does not cure the factual deficiencies of MathWorks, the Official Notice taken, and Alpern in regards to claims 1 and 25. GNU gprof is a profiler used to determine which parts of a program are taking the most execution time (GNU gprof at 1). GNU gprof is used to profile programs, not models including blocks. Thus, GNU gprof does not disclose or suggest ***a debugger interfaced with a model view of a model being executed***, said model comprising ***a block***. GNU gprof also does not disclose ***block methods***, a feature also missing from MathWorks, the Official Notice taken, and Alpern. GNU gprof also does not disclose a debugger that is capable operating at the block level. Therefore, GNU gprof does not remedy the shortcomings of MathWorks with respect to at least the above-mentioned features of claims 1 and 25.

For at least the reasons presented above, MathWorks, the Official Notice taken, Alpern, and GNU gprof, alone or in any reasonable combination, do not disclose each and every element of independent claim 1. Claims 4, 10-16, and 23depend from claim 1 and, as such, include each and every element of claim 1. Claims 28, 34-40, and 47 depend from claim 25 and, as such, include each and every element of claim 25. Therefore, MathWorks, the Official Notice taken, Alpern, and GNU gprof do not disclose each and every element of claims 4, 10-16, 23, 28, 34-40, and 47. Applicants therefore respectfully request that the 35 U.S.C. §103(a) rejection of claims 4, 10-16, 23, 28, 34-40, and 47 be withdrawn.

CONCLUSION

Applicants believe the pending application is in condition for allowance. If the Examiner feels that further discussion would expedite the proceedings, the Examiner is urged to call Applicants' attorney at the phone number listed below.

Please charge any shortage or credit any overpayment of fees to our Deposit Account No. 12-0080, under Order No. MWS-106RCE. In the event that a petition for an extension of time is required to be submitted herewith, and the requisite petition does not accompany this response, the undersigned hereby petitions under 37 C.F.R. §1.136(a) for an extension of time for as many months as are required to render this submission timely. Any fee due is authorized to be charged to the aforementioned Deposit Account.

Dated: June 29, 2009

Respectfully submitted,

Electronic signature: /Kevin J. Canning/
Kevin J. Canning
Registration No.: 35,470
LAHIVE & COCKFIELD, LLP
One Post Office Square
Boston, Massachusetts 02109-2127
(617) 227-7400
(617) 742-4214 (Fax)
Attorney/Agent For Applicant